Assignment Report

Performance Stats.

Sayed Kotb Sayed

28th September, 2018

Selection Sort

1000_unsorted: 1.76032
1000_sorted: 1.64569

5000_unsorted: 40.2838
5000_sorted: 37.9107

10000_unsorted: 166.444
10000_sorted: 152.245

50000_unsorted: 3881.14
50000_sorted: 3856.17

75000_unsorted: 8873.48
75000_sorted: 8678.19

100000_unsorted: 15541.9
100000_sorted: 15488.9

500000_unsorted: 400517

500000_sorted: 395923

Comments:
1- We can notice the quadratic growth in the execution time.
2- We can also notice the data independence as the time take to sort the unsorted or the sorted datasets is almost equal.

Insertion Sort

1000_unsorted: 1.02113
1000_sorted: 0.006934

5000_unsorted: 24.5629
5000_sorted: 0.033244

10000_unsorted: 89.3198
10000_sorted: 0.065214

50000_unsorted: 1962.56
50000_sorted: 0.234192

75000_unsorted: 4449.39
75000_sorted: 0.320534

100000_unsorted: 7903.48
100000_sorted: 0.424844

500000_unsorted: 211913

500000_sorted: 2.14973

Comments:
1- We can notice the quadratic growth in the execution time.
2- We can also notice the data dependence as the time take to sort the sorted dataset is much less than the time taken to sort the unsorted dataset, almost constant (Best case).

Merge Sort

1000_unsorted: 0.797598
1000_sorted: 0.466235

5000_unsorted: 3.88383
5000_sorted: 2.51888

10000_unsorted: 8.17525
10000_sorted: 7.41998

50000_unsorted: 38.2267
50000_sorted: 25.8856

75000_unsorted: 55.2401
75000_sorted: 38.8808

100000_unsorted: 73.5054
100000_sorted: 52.0414

500000_unsorted: 366.693

500000_sorted: 282.72

Comments:
1- We can notice the n.log(n) growth in the execution time.
2- We can also notice the data independence as the time take to sort the unsorted or the sorted datasets is almost equal.

Randomized Quick Sort

1000_unsorted: 0.364578
1000_sorted: 0.249253

5000_unsorted: 1.4872
5000_sorted: 1.23917

10000_unsorted: 3.64819
10000_sorted: 2.46441

50000_unsorted: 17.8715
50000_sorted: 13.2078

75000_unsorted: 29.828
75000_sorted: 21.0264

100000_unsorted: 37.764
100000_sorted: 27.7143

500000_unsorted: 202.627

500000_sorted: 154.827

Comments:
1- We can notice the n.log(n) growth in the execution time.
2- We can also notice the data dependence as the time take to sort the sorted dataset is much higher than the time taken to sort the unsorted dataset, almost quadratic (Worst case).

Heap Sort

1000_unsorted: 0.45886
1000_sorted: 0.320742

5000_unsorted: 1.8786
5000_sorted: 1.65086

10000_unsorted: 4.07218
10000_sorted: 3.68997

50000_unsorted: 23.2341
50000_sorted: 27.5959

75000_unsorted: 39.3381
75000_sorted: 31.4483

100000_unsorted: 50.0789
100000_sorted: 42.8571

500000_unsorted: 320.093

500000_sorted: 252.768

Comments:
1- We can notice the n.log(n) growth in the execution time.
2- We can also notice the data independence as the time take to sort the unsorted or the sorted datasets is almost equal.

Hybrid Sort (TimSort)

1000_unsorted: 0.260012
1000_sorted: 0.093923

5000_unsorted: 1.43507
5000_sorted: 0.758824

10000_unsorted: 2.74604
10000_sorted: 1.92994

50000_unsorted: 17.4421
50000_sorted: 9.22397

75000_unsorted: 32.8778
75000_sorted: 14.6601

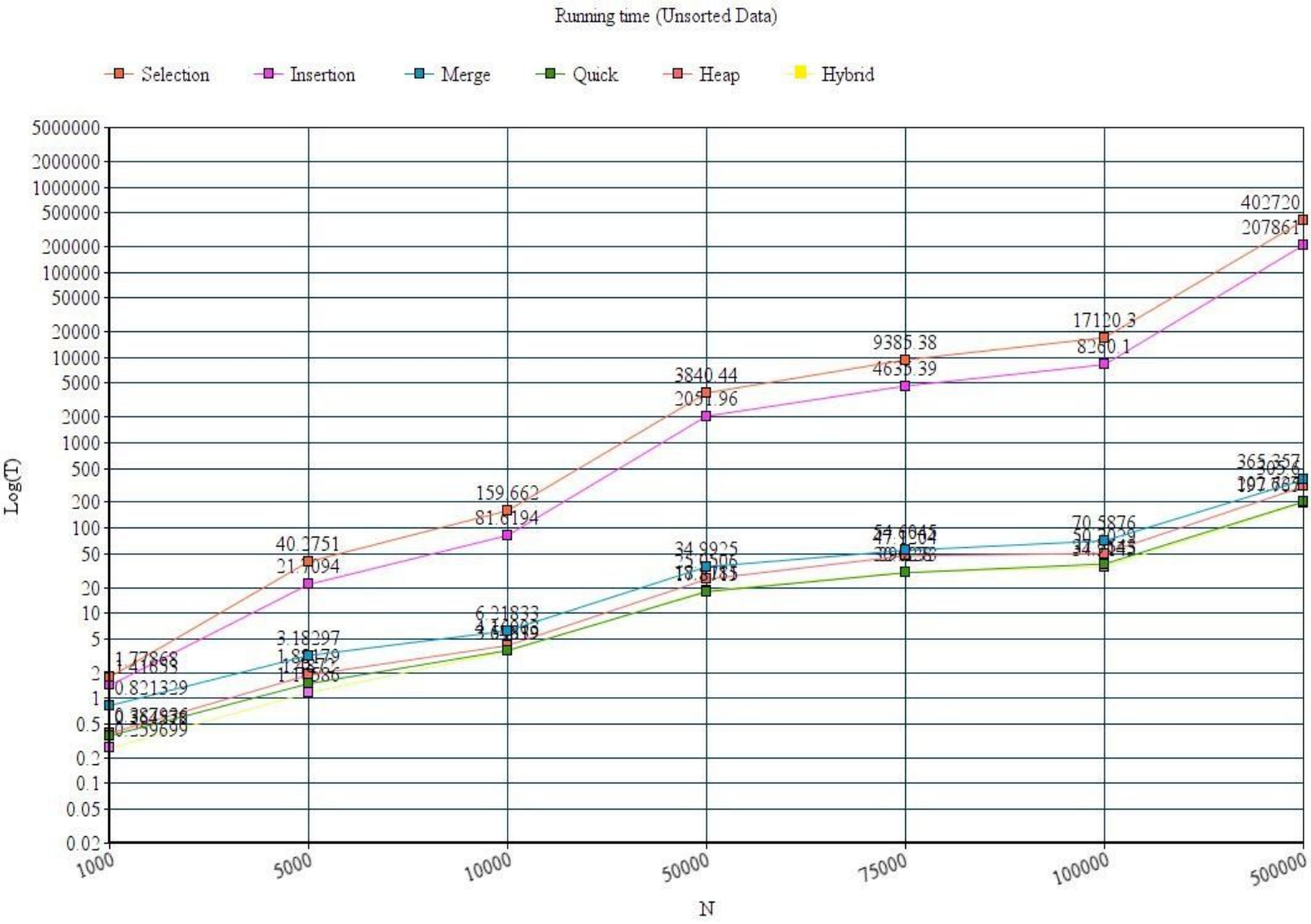
100000_unsorted: 37.8129
100000_sorted: 20.0686

500000_unsorted: 223.18

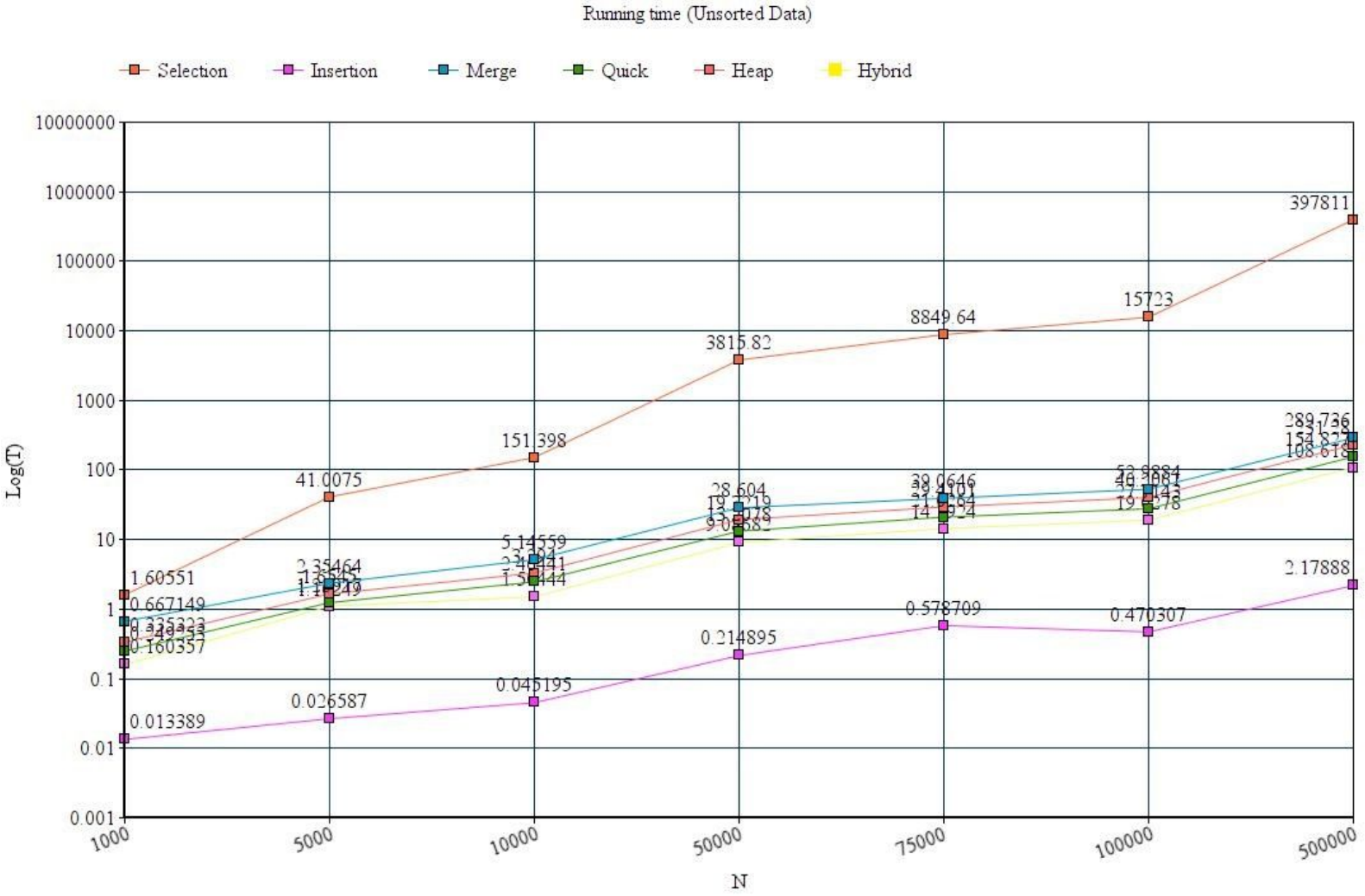
500000_sorted: 113.956

Comments:
1- We can notice the n.log(n) growth in the execution time.
2- We can also notice the data independence as the time take to sort the unsorted or the sorted datasets is almost equal.
3- We can notice about 40% improvement in the runtime compared to the conventional Merge Sort.

Plot (Unsorted)



Plot (Sorted)


Running time (Unsorted Data)
Selection
Insertion
Merge
Quick
Heap
Hybrid
Log(T)
N
10000000
1000000
100000
10000
1000
100
10
1
0.1
0.01
0.001
1000
5000
10000
50000
75000
100000
500000
397811
15723
8849.64
3815.82
151.398
41.0075
1.60551
0.667149
0.160357
0.013389
0.026587
0.045195
0.214895
0.578709
0.470307
2.17888